Connor Elkhill

Module 2 Day 3 Homework

**XXXXXXXXXXXXXXXXXXXXXXXXXZZZZZZZZZZZZYYYYYYYYYYYY**

Input Query Sequence:

Input Sequencer Reads:

**ZZZZWZYZYYY**

**XXXXZZZZZ**

**ZZYYYY**

**ZZZZZ**

**XXXXXXX**

**YYYYYY**

**BBBB**

**BBBBXXZZZ**

**YYYYYVY**

**…**

**ZZZZZZZZZZZZ**

Output Contig Containing Initial Query:

**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

**9**

**10**

**DDDDD**

In order to find out how my program would handle these reads, I ran the reads and the query sequence through my program where the input reads are numbered according to the figure above. Here are my results:

>FinalContig:Length:48

BBBBBBXXXXXXXXXZZZZZZZZZZZZZZZZZZZZYYYYYYYYYYYYV

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sseqid | sstart | send | qstart | qend |
| >4 | 15 | 20 | 0 | 5 |
| >4 | 20 | 25 | 0 | 5 |
| >4 | 25 | 30 | 0 | 5 |
| >4 | 30 | 35 | 0 | 5 |
| >3 | 11 | 20 |  |  |
| >5 | 6 | 13 |  |  |
| >6 | 33 | 39 |  |  |
| >7 | 35 | 41 |  |  |
| >7 | 41 | 47 |  |  |
| >8 | 0 | 4 |  |  |

The contig that my program produced contains 6 different reads from this initial set of reads that were given and is only 1 nucleotide shorter than the output contig containing the initial in the example given. A few notes that I would like to make about this example and how it relates to my implementation:

1. **Reverse complement:** My implementation currently does not allow reads to be the reverse complement. The initial implementation of just forwards reads made this problem more feasible from the start and is something I expect to include in the future. If we allow for the reverse complement (and assume that X and Y are complements in this example) I believe my program would arrive at the same output as the example.

2. **Error correction:** My current implementation also does not correct errors found in the reads. In the example, there is likely an error in read 1, where the V should actual be a Y. It is more common to see errors later in the read, therefore an error at this position makes more sense than an error at position 1 or 2 of the read. Since these input reads are short, we expect only a few errors at most in the read as a whole, so they could potentially be corrected. Errors in De Bruijn Graphs appear as “bubbles” from the rest of the graph, often with nodes that have very low degrees. A way to correct for these errors is to examine the graph for “weak” kmers, which are nodes with low degrees. Then, the error can be corrected by comparing it to nodes that have close sequence Hamming distance but have a much stronger degree.